UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Faculty of Information and Communication Technology

ALGORITHM-PROGRAM VISUALIZATION MODEL: AN INTEGRATED SOFTWARE VISUALIZATION TO SUPPORT NOVICES' PROGRAMMING COMPREHENSION

Affandy

Doctor of Philosophy

2015

ALGORITHM-PROGRAM VISUALIZATION MODEL: AN INTEGRATED SOFTWARE VISUALIZATION TO SUPPORT NOVICE'S PROGRAMMING COMPREHENSION

AFFANDY

A thesis submitted
in fulfilment of the requirements for the degree of Doctor of Philosophy

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2015

DECLARATION

I declare that this thesis entitled “Algorithm-Program Visualization Model: An Integrated Software Visualization to Support Novices’ Programming Comprehension“ is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……

Name : Affandy

Date : ……

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Doctor of Philosophy

Signature	:	……
Supervisor Name	:	Prof. Dr. Nanna Suryana Herman
Date	:	……

DEDICATION

In the name of Allah, the most beneficent, the most merciful

Al-hamdulillahi rabbil ‘alamin

All the praises and thanks be to Allah, the Lord of the 'Alamin

this work is dedicated to:

my beloved parents
[H. Achmad Rozy & almarhumah Hj. Djumirah]

my endless loves, my wife and my daugther
[Trabalista Purwaningrum & Nobelia Salma Aisyahfira]

my siblings and the families
[Heni Isnaeni, Iin Indriani, and Roziqin]

my parents-in-law and the families
[Heru Sutrisno & Mamiek, Yulianto, Nuning, and Tatik]

ABSTRACT

Computer programming is the essential foundation for the other basic skills in Information Technology knowledge areas. Success in this field requires complex knowledge and skill. Mostly, conventional programming courses have been delivered based on the programming textbooks with professional developer tools which focus on the syntax or semantic through the coding task. The role of Software Visualization (SV) has been involved to overcome the complexity and problems in the learning programming. It represents the abstractness of the program in graphical views or illustrations of its entities. Nevertheless, the outcome of the learning still remains poor. Through multi-methodological approach, this research aimed to improve the effectiveness of the visualization as the program comprehension tool. It is found that the interrelated tasks in the programming process, with its various abstractions, and timing in delivering the feedback, need to be addressed with the equal attention in learning to program. Taking into account from those main issues, this study introduces the new model of integrated algorithm-program visualization (ALPROV) for developing program comprehension tool. This model is then to be used in the prototype tool development that is called 3De-ALPROV (Design Development Debug – Algorithm Program Visualization). The efficacy evaluation of the prototype is based on pre- and post- test of the students' programming performance. The programming performances from the treatment and control group are compared to analyze the effect of using the proposed tool in learning programming. Respondents are first-year bachelor students who lack of programming knowledge and experience.Analysis proved that using the program comprehension tool, which has been developed using integrated ALPROV model significantly improved the treatment group's programming performance. Conducting other experiments as the extended study, such as seek for a larger group of respondents, conduct the experiments throughout the necessary period, and use various methods for programming assessment and analysis may improve the findings of this research.

ABSTRAK

Pengaturcaraan komputer adalah asas penting bagi kemahiran yang lain dalam bidang teknologi maklumat. Kejayaan di bidang ini memerlukan pengetahuan dan kemahiran yang kompleks. Kebanyakan kursus pengaturcaraan konvensional disampaikan berdasarkan buku teks pengaturcaraan dengan alat pembangun profesional yang tertumpu kepada syntak/semantik melalui tugasan penulisan program. Visualisasi Perisian (SV) telah dibabitkan bagi mengatasi kerumitan dan masalah dalam pembelajaran pengaturcaraan. Ianya mendedahkan keabstrakan program dalam bentuk grafik atau ilustrasi dari entitinya. Walau bagaimanapun, masih terdapat kekurangan dalam hasil pembelajaran. Melalui pendekatan multi-methodological, matlamat kajian ini untuk meningkatkan keberkesanan visualisasi sebagai alat pemahaman program. Kajian ini mendapati bahawa tugas-tugas yang saling berkait dalam proses pengaturcaraan dengan pelbagai abstraksi program dan penyampaian maklum balas, perlu ditangani dengan perhatian yang sama masa belajar membuat program. Mengambil kira dari isu utama ini, kajian ini memperkenalkan model baru dari algoritma-program visualisasi bersepadu (ALPROV) untuk membangunkan alat pemahaman program. Model ini digunakan dalam pembangunan prototaip yang dipanggil dengan nama 3DE-ALPROV (Rekabentuk Pembangunan Debug - Algoritma Program Visualisasi). Penilaian keberkesanan prototaip adalah berasaskan kepada peperiksaan pra dan pasca pencapaian pengaturcaraan pelajar. Pencapaian pengaturcaraan daripada kumpulan perlakuan dan kawalan, dibandingkan dengan memerhati, mengukur, dan menganalisis keberkesanan penggunaan alat visualisasi yang dicadangkan dalam mempelajari pengaturcaraan. Responden adalah pelajar sarjana muda tahun pertama yang kurang pengetahuan dan pengalaman dalam pengaturcaraan.Analisis membuktikan bahawa menggunakan alat pemahaman pengaturcaraan yang telah dibangunkan dengan menggunakan model ALPROV bersepadu dapat meningkatkan pencapaian pengaturcaraan dari pada kumpulan perlakuan dengan signifikan. Menjalankan eksperimen yang lain sebagai kajian lanjutan, iaitu mendapatkan kumpulan responden yang lebih besar, eksperimen dalam tempoh yang cukup, menggunakan pelbagai kaedah untuk penilaian pengaturcaraan dan analisis boleh menambah baik kepada hasil kajian ini.

ACKNOWLEDGEMENTS

It is a privilege and a great pleasure to thank to many people who made this thesis possible.

First and foremost I would like to express my gratitude and sincere appreciation to my supervisor, Professor Dr. Nanna Suryana, through close interaction for support, encouragement, guidance and critics that continuously influence my way of thinking about research and living the life. The gratitude goes also to my co-supervisor, Associate Professor Dr. Burairah Husin, for his endless source of energy for sharing knowledge, expertise and experiences.

I would like to extend my special appreciation to Dr. Ir. Edi Noersasongko that has provided this opportunity, resources and supports for making it happen. To Dr Abdul Syukur for encourage, support, and motivate me to pursue my PhD. And to all staffs from the Faculty of Information and Communication Technology and the International Office in UTeM and also in UDinus for providing the excellent services and a comfortable environment during my study.

My sincere appreciation also extends to the evaluation committee, Professor Dr. Baharuddin Aris, Dr. Gede Pramudya Ananta, and Assoc. Professor Dr. Burhanuddin Mohd. Aboobaider, for their valuable input and guidance, and to all of my colleagues who have provided assistance at various occasions especially to Dr. Fahmi Arif, Romi Satrio, Basem Zughoul, Daniel Hartono, and Mohamed Doheir.

Heartfelt thanks to my beloved parents, and parents-in-law, my brother, and my sisters for supports, prayer, and understanding. And last but not least for the most precious in my life, my wife Trabalista, and my daughter Nobelia; for your immense support I shall remain forever grateful since without both of you it would all be meaningless.

May the Almighty Allah will always guide you in every step you take, bless you with more than enough, and make things easy for you when life gets rough.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND GLOSSARIES

3De-ALPROV - *Design-Development-Debug Algorithm-Program Visualization.* A prototype of the program comprehension tool that is built based on the integrated model of algorithm-program visualization.

ACM - *Association for Computing Machinery*. The largest and oldest international organization of scientific and educational computer society in the industry.

ALPROV - *Algorithm-Program Visualization.* The new model of visualization tool that emphasizes learning programming on the process of the program and multiple program's abstraction

AV - *Algorithm Visualization*. The visualization of the higher-level abstractions which describe software (Stasko et al., 1998).

BLS - *Bureau of Labour Statistics.* The principal Federal agency in United States that responsible for measuring labour market activity, working conditions, and price changes in the economy.

CCS - *Computing Classification System.* A subject classification system for computing devised by the Association for Computing Machinery (ACM).

CFG - *Control-Flow Graph.* A representation, using graph notation, of all paths that might be traversed through a program during its execution.

CS - *Computer Science.* A wide range of computing discipline that falls into three categories: design and implement software, devise the new way to use computer, and develop effective ways to solve computing problems.

CSD - *Control Structure Diagram.* A notation for the graphical representation of algorithms in detailed designs as well as actual source code. It is intended to reduce the time required to comprehend software by clearly depicting the control constructs and control flow (James H. Cross et al., 1998).

DSV - *Data Structure Visualization.* An interactive animation and fairly self-explanatory for understanding a variety of data structures and algorithms, developed by David Galles in University of San Fransisco (Galles, 2011).

DV - *Data Visualization.* The presentation of detailed and dynamic views of specified variables, and their structure, within executing code that allows users to observe their behaviour as the execution sequence progresses, providing invaluable debugging information (Ellershaw, Oudshoorn, 1994).

eL-CID - *e-Learning to Communicate Iterative Development.* An XML application system to show students examples of computer programs' iterative development using visualisation techniques(Boisvert, 2009)

IDE - *Integrated Development Environment*, A software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a source code editor, build automation tools and a debugger.

IEEE-CS - *Institute of Electrical and Electronics Engineers-Computer Society.* The computing professional's single, unmatched source for technology - information, inspiration and collaboration. By making the most up-to-date and advanced information in the computing world

IT - *Information Technology*. The technology involved in acquiring, storing ,processing, and distributing information by electronic means, including radio, TV, telephone and computers

ITiCSE - *Innovation and Technology Computer Science Education.* Conference from special interest group on computer science education (SIGCSE) that has been held annually since 1996, in many different countries to address pressing issues in computing education.

JDI	-	*Java Debugger Interface.* A high level Java API providing information useful for debuggers and similar systems needing access to the running state of a (usually remote) virtual machine.
JSP	-	*Jackson Structured Programming. A* method for structured programming based on correspondences between data stream structure and program structure.
MCSA	-	*Multiple-Choices Single-Answer.* A question in which respondents are asked to select the best possible answer out of the choices from a list.
PV	-	*Program Visualization.* The visualization of actual program code or data structures in either static or dynamic form (Stasko et al., 1998)
RAPTOR	-	*Rapid Algorithm Prototyping Tool.* A visual programming environment, designed specifically to help students envision their algorithms and avoid syntactic baggage (Carlisle et al., 2005).
SD	-	*Standard Deviation.* A value that shows how much variation or dispersion from the average exists.
SOLO	-	*Structure of the Observed Learning Outcome,* A model of classifying learning outcomes in terms of their complexity, enabling to assess students' work in terms of its quality.
SV	-	*Software Visualization.* The use of the crafts of typography, graphic design, animation, and cinematography with modern human-computer interaction and computer graphics technology to facilitate both the human understanding and effective use of computer software (Stasko et al., 1998).
SWEBOK	-	*Software Engineering Body of Knowledge.* A baseline for the body of knowledge for the field of software engineering that have been established by IEEE Computer Society.
UDiNus	-	*Universitas Dian Nuswantoro*
UTeM	-	*Universiti Teknikal Malaysia Melaka*
VP	-	*Visual Programming.* The use of "visual" techniques to specify or manipulating program elements graphically with visual expression, spatial arrangement and graphic symbols.
WYSIWYC	-	*What You See Is What You Code.* A "radically dynamic" development model to facilitate learner-constructed algorithm visualizations in which the line of algorithm code currently being edited is re-evaluated on every edit, leading to the dynamic update of an accompanying visualization of the algorithm (Hundhausen, Brown, 2007).

LIST OF PUBLICATIONS

Affandy, Suryana, N., & Husin, B. (2014). The Integrated Software Visualization Model to Support Novice's Program Comprehension, *Advanced Science Letters*, *20*(12), 2166-2170.

Affandy, Suryana, N., & Husin, B. (2014). Effectiveness of Integrated Algorithm-Program Visualization: A Case Study with the 3De-AlProV. *Advanced Science Letters*, *20*(1), 304–308.

Affandy, & Suryana, N. (2012). Integrated Algorithm-Program Visualization: a Novel Approach of Software Visualization Development. In *Mobile Learning, Applications, and Services (mobilcase 2012).*

Affandy, Suryana, N., Salam, S., & Azmi, M. (2011). The Development of Synergetic Programming Visualization Tool For Novice Programmers, Short-term Project Report, PJP/2010/FTMK(12E)-S720,FTMK-UTeM.

Affandy, Suryana, N., Salam, S., & Azmi, M. (2011). 3De-synergetic Program Visualization: A visual learning-aid tool for novice students. In *e-Education, Entertainment and e-Management (ICEEE), 2011 International Conference on* (pp. 133–137).

Affandy, Herman, N. S., Salam, S., & Noersasongko, E. (2011). A Study of Tracing and Writing Performance of Novice Students in Introductory Programming. In J. M. Zain, W. M. bt Wan Mohd, & E. El-Qawasmeh (Eds.), *Software Engineering and Computer Systems* (pp. 557–570). Springer-Verlag Berlin Heidelberg.